

Литература

1. IrDA датчик [Электронный ресурс] / Сайт ixbt.com. – Режим доступа: <https://www.ixbt.com/peripheral/irda.html>. – Дата доступа: 21.03.2022.

2. Логический анализатор [Электронный ресурс] / Официальный сайт компании «Контрольно-измерительные приборы и системы». – Режим доступа: https://www.kipis.ru/info/index.php?ELEMENT_ID=20748. – Дата доступа: 21.03.2022.

А. Р. Графов

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. И. Кучеров**, ст. преподаватель

РАЗРАБОТКА НАБОРОВ ТЕСТОВ ДЛЯ ПРОВЕРКИ ФУНКЦИОНИРОВАНИЯ САЙТОВ

Проект был создан для тестирования правильности работы пользовательского интерфейса веб приложений.

Данный проект дает возможность осуществлять полную проверку работы приложения, статических элементов, веб анимации javascript. Пользователь может тестировать целый ряд элементов страницы: текст, текстовые поля, выпадающие списки, кнопки, чекбоксы (checkboxes), переключатели (radiobuttons). Также пользователь может собирать различные автотесты в тестовые наборы и доставлять их в CI/CD системы, а также наблюдать время выполнения как одного теста, так и тестового набора.

Распечатка подробного отчета, фильтрация, параметризация реализуется с помощью тестового фреймворка pytest.

Оптимизация кода, читабельность и соблюдение принципа DRY достигается использованием фикстур.

В основе архитектуры лежит Page object Model. Эта модель подразумевает использование принципов ООП, в частности наследования. Также в этой модели тесты разделяются по страницам веб приложения, используется принцип «один тест покрывает функционал одной страницы».

В основе любого тестового проекта лежит разработка тестового фреймворка. В тестовом фреймворке создается универсальная основа для создания гибких тестовых наборов. В проекте реализован класс Base, в этом классе описаны все методы, которые могут использоваться

в тестах, в основном методы локализации элементов. Все последующие классы описывают веб-страницы и являются потомками класса Base. Локаторы элементов хранятся в отдельном файле. Фикстуры находятся в файле confTest и отвечают за параметризацию и автоматизацию кода.

Весь проект написан с использованием языка Python, в качестве веб-драйвера используется Selenium, используемая IDE – Pycharm Community Edition.

А. Р. Графов

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. И. Кучеров**, ст. преподаватель

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ SELENIUM WEBDRIVER ПРИ НАПИСАНИИ АВТОМАТИЗИРОВАННЫХ ТЕСТОВ

Selenium поддерживает автоматизацию всех основных браузеров на рынке с помощью WebDriver. WebDriver – это протокол, определяющий независимый от языка интерфейс для управления поведением веб-браузеров. Каждый браузер поддерживается определенной реализацией WebDriver, называемой драйвером. Драйвер – это компонент, который отвечает за передачу полномочий браузеру и обеспечивающий обмен данными между Selenium и браузером.

Это разделение является частью сознательного усилия, направленного на то, чтобы поставщики браузеров взяли на себя ответственность за реализацию своих браузеров. Selenium по возможности использует эти сторонние драйверы, но также предоставляет свои собственные драйверы, поддерживаемые проектом, для тех случаев, когда это невозможно.

Фреймворк Selenium связывает все эти части вместе с помощью пользовательского интерфейса, который позволяет прозрачно использовать различные серверные части браузера, обеспечивая кросс-браузерную и кроссплатформенную автоматизацию.

Настройка Selenium сильно отличается от настройки других коммерческих инструментов. Прежде чем начать писать код Selenium, необходимо установить библиотеки языковых привязок для выбранного вами языка, браузера, который вы хотите использовать, и драйвера для этого браузера.